On Polynomial Time Many-One Completeness of One-Way Functions : Preliminary Report

AUTHOR(S):

WATANABE, Osamu; TODA, Seinosuke

CITATION:

WATANABE, Osamu ...[et al]. On Polynomial Time Many-One Completeness of One-Way Functions : Preliminary Report. 数理解析研究所講究録 1989, 695: 162-168

ISSUE DATE:

1989-06

URL:

http://hdl.handle.net/2433/101393

数理解析研究所講究録
第 695 巻 1989 年 162-168

On Polynomial Time Many-One Completeness of One-Way Functions

(Preliminary Report)

Osamu WATANABE
Dept. Computer Science
Tokyo Institute of Technology
Meguro-ku, Tokyo 152

Seinosuke TODA
Dept. Computer Science
University of Electro-Communications
Chofu-shi, Tokyo 182

1. Introduction

The subject of this paper is to investigate intractability of computing inverses of one-way functions. A function is called "one-way" in general if its inverse is "harder" to compute than the function itself. Recently, one-way functions have received considerable attention because of their practical application as well as their theoretical importance. Note that there are many definitions for "one-way function" (see, e.g., [Wat88]). In this paper we define "one-way" as follows: a *one-way function candidate* is a one-to-one, honest, and polynomial time computable function, and a *one-way function* is a one-way function candidate whose inverse is not polynomial time computable.

Consider any one-way function (candidate) f. Intuitively, polynomial time nondeterministic computation is sufficient for inverting f; indeed, if f^{-1} is not polynomial time computable, then P $\neq$ NP. Thus, it seems very hard to prove the polynomial time non-invertibility of f^{-1}. Structural complexity theory provides several quantitative scales to measure "intractability" of problems. For example, while we have been unable to prove that SAT is not polynomial time solvable, we can prove that SAT is the "hardest" problem in NP, where the "completeness" notion is used to state "hardest". Here we investigate completeness of computing inverse of one-way function candidates.

Note that conventional computational complexity theory has been developed mostly for "decision problems", whereas we are considering problems of evaluating function values; thus, we have to either characterize computation for function evaluations in terms of decision problems or introduce the corresponding notions into our context. Valiant [Va76] defined the class UP in order to characterize computation for inverting one-way function candidates. Since then, several interesting results have been observed

concerning the complexity of UP, e.g., [Wat88]. However, this approach, the study of function evaluation problems in the context of decision problems, has limitation, and thus, some researchers [Be88, Kr86] have started investigating more directly. We follow this latter approach. We consider classes of functions, $\Box_2^{\mathrm{P}}$ and OptP [Kr86], that provide upper bounds for inverting one-way function candidates. We define "$\leq_{\mathrm{m}}^{\mathrm{PF}}$-reduciblity" (or, "metric reducibility" [Kr86]) that corresponds $\leq_{\mathrm{m}}^{\mathrm{P}}$-reducibility, thereby discussing "$\leq_{\mathrm{m}}^{\mathrm{PF}}$-completeness" of inverting one-way function candidates in $\Box_2^{\mathrm{P}}$ and OptP. We show that some structural conjecture implies that the inverse of no one-way function candidate is $\leq_{\mathrm{m}}^{\mathrm{PF}}$-complete in $\Box_2^{\mathrm{P}}$ (resp., OptP).

The class P^{NP} $(= \Delta_2^{\mathrm{P}})$ characterizes computation for "optimization problems". Recently some researchers [PZ82, Kr86, He87] have introduced subclasses of P^{NP} by restricting the way and/or the number of queries to oracle sets. We can observe the structure of P^{NP} using these subclasses. Classes $\mathrm{P}_{t(n)\text{-T}}^{\mathrm{NP}}$, $\mathrm{P}_{\mathrm{tt}}^{\mathrm{NP}}$, and L^{NP} are the classes of sets accepted, respectively, by polynomial time oracle machines asking $t(n)$ many queries to some oracle set in NP, by polynomial time oracle machines asking non-adaptive (i.e., truth-table-type) queries to some oracle set in NP, and by log space oracle machines relative to some oracle set in NP. In [BH88, He87, Wag88] it is reported that $\mathrm{P}_{O(\log n)\text{-T}}^{\mathrm{NP}} = \mathrm{P}_{\mathrm{tt}}^{\mathrm{NP}} = \mathrm{L}^{\mathrm{NP}}$. On the other hand, Kadin [Ka87] proved that $\mathrm{P}_{k\text{-T}}^{\mathrm{NP}} \subsetneq \mathrm{P}_{(k+1)\text{-T}}^{\mathrm{NP}}$ if the polynomial time hierarchy collapses to a finite level; thus, it is natural to conjecture the following [Wag86].

Conjecture. $\mathrm{P}_{O(\log n)\text{-T}}^{\mathrm{NP}} \neq \mathrm{P}^{\mathrm{NP}}$.

We prove that this conjecture implies that the inverse of no one-way function candidate is $\leq_{\mathrm{m}}^{\mathrm{PF}}$-complete in $\Box_2^{\mathrm{P}}$ (resp., OptP).

2. Preliminaries

In the following, we define notions and notation that are necessary in this paper. We omit defining standard ones in computational complexity theory: the reader will find them in, e.g., [BDG88].

We use $\Sigma = \{0, 1\}$ for a finite alphabet and assume some natural encoding of the set of integers over Σ. For any string x, we use $|x|$ to denote the length of x. For any oracle

machine M and set A, let $L(M, A)$ denote the set of strings accepted by M relative to A. We assume some tupling function that is polynomial time computable and invertible. We denote the output of the function on $a_1, ..., a_n$ by $\langle a_1, ..., a_n \rangle$. In the following, by a *function* we mean a function from Σ^* to Σ^*; a function is not necessarily total. The composition of two functions f and g is denoted by $f \circ g$. For any y, let $f^{-1}(y)$ denote the set $\{x : f(x) = y\}$. By an *inverse* of f we mean a function g mapping every y in the range of f to some element in $f^{-1}(y)$. Notice that every one-to-one function has a unique inverse. When f is one-to-one, we use f^{-1} also to denote the inverse of f. A function f is *honest* if there exists some polynomial p such that $|x| \leq p(|f(x)|)$ for every x in the domain of f. The "honest" property is necessary to avoid the case that polynomial time non-invertibility is trivial. It is shown [GS88, Va76] that a one-way function exists if and only if P $\neq$ UP ($\subseteq$ NP); we have been unable to find a "real" one-way function.

We use standard notation for complexity classes: e.g., P, NP. Let PF denote the class of polynomial time computable functions. For any oracle set A we write complexity classes relative to A in such a way as P^A. In order to discuss relativized complexity classes in more detail, we use the following notation [BDG88].

Definition 1.

(1) For any set A and any reduction type r, P_r^A is the class of sets that are $\leq_r^{\mathrm{P}}$-reducible to A.

(2) For any set A, $\mathrm{PF}_{\mathrm{T}}^A$ (or, PF^A) is the class of functions that are deterministically polynomial time computable relative to A. A class $\mathrm{PF}_{\mathrm{tt}}^A$ is the class of functions that are deterministically polynomial time computable asking non-adaptive (i.e., truth-table-type) queries to A.

For any class of sets $\mathcal{C}$ and any reduction type r, $\mathrm{P}_r^{\mathcal{C}} = \bigcup_{A \in \mathcal{C}} \mathrm{P}_r^A$ (resp., $\mathrm{PF}_r^{\mathcal{C}} = \bigcup_{A \in \mathcal{C}} \mathrm{PF}_r^A$). We often use conventional notation Π_2^{P} for $\mathrm{PF}^{\mathrm{NP}}$. As mentioned in Introduction, we have the following observation, which plays a key role in this paper.

Proposition 1. [BH88, He87, Wag88] $\mathrm{P}_{\mathrm{tt}}^{\mathrm{NP}} = \mathrm{P}_{O(\log n)\text{-T}}^{\mathrm{NP}}$.

We define the notion of "polynomial time reducibility" for problems of computing

function values. The following is one reasonable definition for "polynomial time many-one reducibility".

Definition 2. For any function f and g, f is *polynomial time many-one reducible* to g ($f \leq_{\mathrm{m}}^{\mathrm{PF}} g$) if there exit polynomial time computable total functions h_1 and h_2 such that $f = \lambda x.h_1(x, h_2 \circ g(x))$.

Remark. This reducibility is called "metric reducibility" in [Kr86].

We define the notions of "hardness" and "completeness" by the analogy of the ones for decision problems.

3. Results

We discuss difficulty of computing inverses of one-way function candidates. First consider the class $\square_2^{\mathrm{P}}$. Note that $\square_2^{\mathrm{P}}$ is an upper bound for inverting one-way functions: namely, we have the following proposition.

Proposition 2. For every one-way function candidate f, f^{-1} is in $\square_2^{\mathrm{P}}$.

Here we ask whether it is the case that f^{-1} is "hardest" in $\square_2^{\mathrm{P}}$ for some one-way function f. We show that if $\mathrm{P}_{O(\log n)\text{-T}}^{\mathrm{NP}} \neq \mathrm{P}^{\mathrm{NP}}$, then the inverse of no one-way function candidate is $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete in $\square_2^{\mathrm{P}}$. That is, the conjecture concerning the structure of P^{NP} implies that the inverse of no one-way function is "hardest" in $\square_2^{\mathrm{P}}$.

For any polynomial time deterministic oracle machine M and any oracle set A, we define a function $f_{\text{PATH-}M^A}$ as follows:

for every $x \in \Sigma^*$, $f_{\text{PATH-}M^A}(x) = \langle a_1, a_2, ..., a_m \rangle$,
where M on x asks queries $q_1, ..., q_m$ to A, and for every i, $1 \leq i \leq m$,
a_i is the answer to the ith query from the oracle A.

Notice that $f_{\text{PATH-}M^A}$ is in $\square_2^{\mathrm{P}}$.

Proposition 3. For any polynomial time deterministic oracle machine M and any oracle set A, $f_{\text{PATH-}M^A}$ is in $\square_2^{\mathrm{P}}$.

Lemma 4. For any $L \in \mathrm{P}^{\mathrm{NP}}$, let M and A be a polynomial time deterministic oracle machine and an oracle set in NP such that $L = L(M, A)$. If $f_{\text{PATH-}M^A} \leq_{\mathrm{m}}^{\mathrm{PF}} f^{-1}$ for some one-way function candidate f, then $L \in \mathrm{P}_{\mathrm{tt}}^{\mathrm{NP}}$.

Proof. The proof is almost immediate from the claim below. □

Claim. For every one-way function candidate f, $f^{-1} \in \mathrm{PF}_{\mathrm{tt}}^{\mathrm{NP}}$.

Remark. Indeed, we can prove that $f^{-1} \in \mathrm{PF}_{\mathrm{tt}}^{\mathrm{UP}}$.

Sketch of Proof Define $Bit(f^{-1})$ by

$$Bit(f^{-1}) = \{\langle x, i, b\rangle : \text{the } i\text{th bit of } f^{-1}(x) \text{ is } b\}.$$

It follows from one-to-one-ness of f^{-1} that $Bit(f^{-1})$ is in UP ($\subseteq$ NP). From one-to-one-ness of f^{-1}, we also have that for every $x \in \Sigma^*$,

$$f^{-1}(x) = b_1 ... b_m \ \leftrightarrow \ \wedge_{1 \le i \le m}(\ \langle x, i, b_i\rangle \in Bit(f^{-1}) \ ).$$

Thus, one can compute $f^{-1}(x)$ asking $\langle x, 1, 0\rangle$, ..., $\langle x, m, 0\rangle$ to $Bit(f^{-1})$. □ CLAIM

Now the following theorem is immediate from Proposition 1, 2, and 3, and Lemma 4.

Theorem 5. Let f be any one-way function candidate. If $\mathrm{P}_{O(\log n)\text{-T}}^{\mathrm{NP}} \neq \mathrm{P}^{\mathrm{NP}}$, then f^{-1} is not $\leq_{\mathrm{m}}^{\mathrm{PF}}$-complete in $\square_2^{\mathrm{P}}$.

Krentel [Kr86] introduced the class OptP in order to study NP optimization problems. Next we consider this class; we obtain a yet stronger result than Theorem 5.

Definition 2.

(1) A metric machine N is a polynomial time bounded nondeterministic Turing machine such that every nondeterministic path writes a string and halts with an accepting/rejecting state. For every string x, $opt^N(x)$ is the lexicographically largest string on any accepting path of N on input x; $opt^N(x)$ is undefined if no accepting path exists.

(2) A function f is in OptP if there is a metric machine N such that $f(x) = opt^N(x)$ for all $x \in \Sigma^*$.

Remark. This definition slightly differs from the original one.

The following function f_{MAXSAT} is one of the typical functions in OptP:

for every $x \in \Sigma^*$, $f_{\mathrm{MAXSAT}}(x) = b_1 b_2 ... b_m$,

where $b_1, ..., b_m$ is the lexicographically maximum satisfying assignment of x.

(*NOTE:* If x is not a formula nor satisfiable, then $f_{\mathrm{MAXSAT}}(x) = 00...0$.)

Proposition 6. [Kr86] f_{MAXSAT} is $\leq_{\text{m}}^{\text{PF}}$-complete in OptP.

It is easy to show that OptP is an upper bound for inverting one-way function candidates.

Proposition 7. For every one-way function candidates f, f^{-1} is in OptP.

It is clear that OptP is a subclass of $\square_2^{\text{P}}$; furthermore, we have the following close relationship between $\square_2^{\text{P}}$ and OptP.

Proposition 8. [Kr86] For every function f, f is in $\square_2^{\text{P}}$ if and only if f is $\leq_{\text{m}}^{\text{PF}}$-reducible to some g in OptP.

Hence, the following theorem follows from Proposition 7 and 8, and Theorem 5.

Theorem 9. Let f be any one-way function candidate. If $\text{P}^{\text{NP}}_{O(\log n)\text{-T}} \neq \text{P}^{\text{NP}}$, then f^{-1} is not $\leq_{\text{m}}^{\text{PF}}$-complete in OptP.

Remark. Note that Theorem 5 is an immediate corollary of this theorem.

References

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró, Structural Complexity I, EATCS Monographs on Theoretical Computer Science, Berlin, Springer-Verlang (1988).

[Be88] R. Beigel, NP-hard sets are P-supserterse unless R = NP, Technical Report 88-4, Dept. Computer Science, The Johns Hopkins University (1988).

[BH88] S. Buss and L. Hay, On truth-table reducibility to SAT and the difference hierarchy over NP, in "Proc. 3rd Conference on Structure in Complexity Theory", IEEE (1988), 224-235.

[GS88] J. Grollmann and A. Selman, Complexity measures for public-key cryptosystems, SIAM J. Comput., 17 (1988) 309-335.

[He87] L. Hemachandra, The strong exponential hierarchy collapses, in "Proc. 19th ACM Ann. Sympos. on Theory of Computing", ACM (1987), 110-122.

[Ka87] J. Kadin, $\text{P}^{\text{NP}}[\log n]$ and sparse Turing-complete sets for NP, in "Proc. 2nd Conference on Structure in Complexity Theory", IEEE (1987), 33-40.

[Ko85] K. Ko, On some natural complete operators, Theoret. Comput. Sci., 37 (1985), 1-30.

[Kr86] M. Krentel, The complexity of optimization problems, in "Proc. 18th ACM Ann. Sympos. on Theory of Computing", ACM (1986), 69-76.

[PZ82] C. Papadimitriou and S. Zachos, Two remarks on the power of counting, in "Proc. 6th GI Conference on Theoretical Computer Science", Lecture Notes in Computer Science 145, Berlin, Springer-Verlag (1983), 269-276.

[Va76] L. Valiant, Relative complexity of checking and evaluating, Inform. Process. Lett., 5 (1976), 20-23.

[Wag86] K. Wagner, More complicated questions about maxima and minima, and some closures of NP, Theoret. Comput. Sci., 51 (1987), 53-80.

[Wag89] K. Wagner, On restricting the access to an NP-oracle, in "Proc. 15th International Colloquium on Automata, Languages and Programming", Lecture Notes in Computer Science 317, Berlin, Springer-Verlag (1988), 682-696.

[Wat88] O. Watanabe, On one-way functions, in "Proc. The International Symposium on Combinatorial Optimization" (1988), to appear.